

CLAIMS

Having thus described our invention, what we claim as new, and desire to secure by Letters Patent is:

1 1. A workload manager for servicing a work
2 request in a distributed computing environment, the
3 workload manager comprising:

4 a management module residing in one or more nodes
5 comprising a cluster of workstations, the management
6 module being responsive to a work request received to
7 transmit a first signal to initiate an execution of a
8 work associated with the work request;

9 an execution module residing in one or more nodes
10 comprising the cluster of workstations, the execution
11 module being responsive to the signal to execute the
12 work, the execution module transmitting a second signal
13 to the management module indicating completion of the
14 execution of the work.

1 2. The workload manager as claimed in claim 1,
2 wherein the management module includes:

3 a classifier being responsive to a work request
4 received to assign the work associated with the work
5 request;

6 one or more routers responsible for distributing a
7 class of work, said one or more routers responsive to the
8 work assigned by the classifier transmitting the first
9 signal to initiate an execution of the work.

1 3. The workload manager as claimed in claim 2,
2 wherein the management module further includes:

3 a policy manager responsive to a current workload
4 handled by the classifier and said one or more routers to
5 generate additional instances of the classifier and said
6 one or more routers to handle the current workload.

1 4. The workload manager as claimed in claim 3,
2 wherein the execution module includes:
3 one or more initiators being responsive to the
4 first signal to execute the work, said one or more
5 initiators transmitting the second signal to said one or
6 more routers indicating completion of the execution of
7 the work.

1 5. The workload manager as claimed in claim 4,
2 wherein said execution module further includes:
3 one or more objects instantiated by said one or
4 more initiators to execute the work, said objects having
5 same names.

6 6. The workload manager as claimed in claim 5,
7 wherein said objects having same names have different
8 implementations of said objects.

1 7. The workload manager as claimed in claim 4,
2 wherein said one or more initiators reside in one or more
3 nodes in a computer network.

1 8. The workload manager as claimed in claim 7,
2 wherein said one or more nodes include heterogeneous
3 nodes.

1 9. The workload manager as claimed in claim 7,
2 wherein the execution module generates statistics related
3 to the execution of the work, the execution module
4 transmitting the generated statistics to the management
5 module.

1 10. The workload manager as claimed in claim 9,
2 wherein said statistics include memory paging information
3 associated with the work executing in said one or more
4 nodes.

1 11. The workload manager as claimed in claim 9,
2 wherein the statistics include central processing unit
3 (CPU) utilization of said one or more nodes.

1 12. The workload manager as claimed in claim 9,
2 wherein the statistics include input/output (I/O)
3 utilization of said one or more nodes.

1 13. The workload manager as claimed in claim 9,
2 wherein said one or more routers initiates the work on
3 first one of said one or more initiators, said first one
4 determined from said statistics to be best suited to
5 handle the work.

1 14. The workload manager as claimed in claim 4,
2 wherein the execution module further includes:
3 one or more node managers responsive to a request
4 from the first signal to start and stop said one or more
5 initiators.

1 15. The workload manager as claimed in claim 2,
2 wherein the classifier assigns the work to said one or
3 more routers determined to best handle the work.

1 16. The workload manager as claimed in claim 15,
2 wherein the classifier further divides the work into a
3 plurality of subgroups of work and assigns the plurality
4 of subgroups to said one or more routers determined to
5 best handle the plurality of subgroups.

1 17. The workload manager as claimed in claim 2,
2 wherein said one or more routers further include a
3 plurality of router queues to enable the work at
4 different phases of completion to be queued to eliminate
5 potential bottleneck in the workload manager.

1 18. A method of distributing work through a
2 cluster of workstations for efficient distributed
3 processing, said cluster having a plurality of
4 workstations interconnected over a network, the method
5 comprising:
6 receiving a work request;
7 distributing one or more tasks related to the work
8 request over a cluster of workstations;
9 instantiating one or more objects having same
10 class names to execute said one or more tasks; and
11 notifying completion of said one or more tasks.

1 19. The method of distributing work through a
2 cluster of workstations as claimed in claim 18, the step
3 of instantiating includes:

4 instantiating one or more objects having same
5 class names but different implementations to execute said
6 one or more tasks.

1 20. The method of distributing work through a
2 cluster of workstations as claimed in claim 18, the
3 method further including:

4 collecting statistics relating to the step of
5 distributing and execution of said one or more tasks.

1 21. The method of distributing work through a
2 cluster of workstations as claimed in claim 20, wherein
3 the said statistics collected include:

4 hardware specific information related to a node
5 executing said one or more tasks.

1 22. The method of distributing work through a
2 cluster of workstations as claimed in claim 20, wherein
3 the said statistics collected include:

4 operating system specific information related to a
5 node executing said one or more tasks.

1 23. The method of distributing work through a
2 cluster of workstations as claimed in claim 20, wherein
3 the said statistics collected include:

4 performance statistics related to a node executing
5 said one or more tasks.

1 24. The method of distributing work through a
2 cluster of workstations as claimed in claim 20, the
3 method further including:

4 determining from said collected statistics a best
5 suited node in the cluster of workstations for handling
6 said one or more tasks.

1 25. The method of distributing work through a
2 cluster of workstations as claimed in claim 24, wherein
3 the step of distributing includes:

4 distributing said one or more tasks to said best
5 suited node to handle said one or more tasks.

1 26. The method of distributing work through a
2 cluster of workstations as claimed in claim 18, the
3 method further including:

4 dynamically adding a new node to the cluster of
5 workstations.

1 27. The method of distributing work through a
2 cluster of workstations as claimed in claim 18, the
3 method further including:

4 dynamically removing an old node from the cluster
5 of workstations.

1 28. The method of distributing work through a
2 cluster of workstations as claimed in claim 26, the
3 method further including:

4 receiving a broadcast of availability from a new
5 node to join the cluster of workstations; and the step of
6 dynamically adding includes dynamically adding said new
7 node to the cluster of workstations.

1 29. A method of distributing work through a
2 cluster of workstations for efficient distributed

3 processing, said cluster having a plurality of
4 workstations interconnected over a network, the method
5 comprising:

6 receiving a work request;
7 classifying the work request into one or more
8 tasks;
9 assigning said one or more tasks to one or more
10 router queues capable of handling said one or more tasks;
11 dispatching said assigned one or more tasks for
12 execution; and
13 determining performance statistics associated with
14 said one or more router queues.

1 30. The method of distributing work through a
2 cluster of workstations as claimed in claim 29, the
3 method further including:

4 computing a time lapse between the step of
5 assigning and the step of dispatching.

1 31. The method of distributing work through a
2 cluster of workstations as claimed in claim 29, wherein
3 the step of dispatching includes:

4 determining one or more initiators best suited to
5 execute said one or more tasks; and

6 dispatching said one or more tasks to said best
7 suited one or more initiators for execution.

1 32. The method of distributing work through a
2 cluster of workstations as claimed in claim 31, wherein
3 the method further includes:

4 adding additional initiators to execute said one
5 or more tasks based on the performance statistics of said
6 one or more router queues.

1 33. The method of distributing work through a
2 cluster of workstations as claimed in claim 31, wherein
3 the step of determining further includes:

4 receiving from said one or more initiators system
5 specific statistics data associated with said one or more
6 initiators for determining said one or more initiators
7 best suited to execute said one or more tasks.

1 34. The method of distributing work through a
2 cluster of workstations as claimed in claim 31, wherein
3 the step of determining further includes:

4 computing performance statistics of said one or
5 more initiators for determining said one or more
6 initiator best suited to execute said one or more tasks.

1 35. The method of distributing work through a
2 cluster of workstations as claimed in claim 31, wherein
3 the method further includes:

4 distributing objects with same class name having
5 different implementations over a cluster of workstations
6 which objects said one or more initiators invoke to
7 execute said one or more tasks.

1 36. A program storage device readable by machine,
2 tangibly embodying a program of instructions executable
3 by the machine to perform the method steps of
4 distributing work through a cluster of workstations for

5 efficient distributed processing, the method steps
6 comprising:
7 receiving a work request;
8 distributing one or more tasks related to the work
9 request over a cluster of workstations;
10 instantiating one or more objects having same
11 class names to execute said one or more tasks;
12 collecting statistics relating to the step of
13 distributing and execution of said one or more tasks;
14 notifying completion of said one or more tasks.

1 37. The program storage device as claimed in
2 claim 36, wherein the method step of instantiating
3 includes:

4 instantiating one or more objects having same
5 class names but different implementations to execute said
6 one or more tasks.

1 38. The program storage device as claimed in
2 claim 36, the method steps further including:

3 determining from said collected statistics a best
4 suited node in the cluster of workstations for handling
5 said one or more tasks.